

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Patent Application of: Sean E. Aschen et al

Group Art Unit: 2168 : IBM Corporation
Examiner: Olubusola Oni : Intellectual Property Law
Serial No.: 10/603,999 : Department IQ0A/040-3
Filed: 06/25/2003 : 1701 North Street
Title: FILE RETRIEVAL : Endicott, New York 13760
METHOD AND SYSTEM

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDED APPEAL BRIEF

Dear Sir:

In response to a Notification of Non-Compliant Appeal Brief dated 12/07/2006, Appellants hereby submit this Amended Appeal Brief.

(i) REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, a corporation of New York, with a place of business at Armonk, NY 10504.

(ii) RELATED APPEALS AND INTERFERENCES

The present application is related to application Serial No. 10/604,000 also filed 06/25/2003 for which an Appeal Brief was filed 08/08/2006. An Examiner's answer has not yet been received.

(iii) STATUS OF CLAIMS

Claims 7 - 19 are pending in the present application. Claims 7 - 19 have all been finally rejected and are the subject matter of this appeal.

(iv) STATUS OF AMENDMENTS

An amendment after final was filed 07/28/2006. The Examiner Advisory Action of 08/26/2006 does not indicate whether this amendment will be entered or not.

(v) SUMMARY OF CLAIMED SUBJECT MATTER

Appellants' invention relates to a method and system for providing data stored by a first application in a database

residing on a server to other applications residing on other servers or client computers. Such data is provided to the other applications without the need or use of the first application, and in a simple object access protocol.

According to Appellants' independent claim 7, a mailfile stored on a server, of documents having a section and fields, is provided. Mailfile 12 is shown in FIG. 1 residing on a DOMINO server. Support is also found in Appellants' Specification page 5, line 18, to page 6, line 28. A request for one of the documents is received as a SOAP (simple object access protocol) protocol message from an application 18 running on a user workstation different from the server. See page 7, lines 1 - 8. The fields of the requested document are retrieved from the mailfile. See FIG. 2, step 40, and page 7, lines 27 - 28. In response to the fields, the document is retrieved as a markup language document. See step 42.

At the server, a URL is inserted in the markup language document to retrieve the section of the document. See page 8, lines 5 - 8. The section is retrieved at the server from the mailfile in the markup language. See page 8, lines 7 - 8.

An object is created at the server. The object has the section expanded in the retrieved document. The URL is then removed from the retrieved document at the server. See page 8, lines 8 - 10 and 25 - 26. The object is marshalled and sent as a SOAP protocol message to the requesting application. See page 9, lines 4 - 15.

According to independent claim 16, a system provides data stored in a file to an application. The system includes a file stored on a server having data stored as documents having sections. See page 5, line 18, to page 6, line 9, and page 3, lines 6 - 9. There is a database on the server for passing a request for one of the documents to the file and upon return, converting the document into an extended markup format. See page 7, lines 15 - 22. There is also an authentication directory having authentication records for an application. See page 7, lines 9 - 14.

Web service software is included for receiving a request from an application 18 running on a user workstation different from the server, for one of the documents, retrieving the document, and creating an extended markup object containing the document. See page 7, line 23 - page 8, line 1.

There is also web service software for inserting at the server, a URL into the markup object to retrieve one of the sections. See page 8, lines 5 - 8. Web service software is also present for retrieving at the server, the section from the file, and for removing at the server, the URL from the retrieved document, and for creating an object having the section expanded in the retrieved document. See page 8, lines 8 - 26.

The system also includes a protocol tool for authenticating the application using the (authentication) records, marshalling the object and sending the marshalled object to the application in a simple object access protocol. See page 9, lines 8 - 18, and lines 2 - 3.

(vi) GROUND OF REJECTION

There are two grounds of rejection.

First Ground of Rejection:

Claims 7 - 10 and 12 - 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Benninghoff (U.S. Patent Publication 2002/0091782) and Don (W3C Copyright 2000) and further in view of Grout (U.S. Patent Publication 5,913,033).

Specifically, the Examiner rejects independent claim 7 citing Benninghoff as teaching three of the first four steps of providing and two retrieving clauses. The Examiner cites Grout as teaching the subsequent steps of inserting, retrieving, and removing clauses. Finally, the Examiner cites Don as teaching the receiving step and presumably the marshalling step.

Appellants' other independent claim 16 is rejected on the same grounds.

Appellants disagree that the cited portions of Benninghoff, Don, and Grout completely describe or suggest independent claims 7 and 16 as argued below.

Second Ground of Rejection:

Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Benninghoff, Don, and Grout, and further in view of Little (U.S. Patent Publication 2005/0114671).

Appellants do not argue claim 11 separately. Appellants contend all of the dependent claims 8 - 15 and 17 - 19 are allowable because of their dependence on allowable independent claims 7 or 16.

(vii) ARGUMENT

First Ground of Rejection:

Claims 7 - 19 are patentable under 35 U.S.C. 103(a) over the prior art and particularly, U.S. Patent Publications 2002/0091782 (Benninghoff), 5,913,033 (Grout) and W3C Copyright 2000 (Don).

The combination of Benninghoff, Grout, and Don does not describe or suggest all of the required steps of Appellants' claim 7. Appellants therefore respectfully disagree with this rejection and offer the following arguments in support thereof.

Appellants' claim 7 requires that a document be retrieved in response to the fields which are retrieved from the mailfile stored on the server, in response to the request received for the document which has a section. The server has to insert a URL into the document to retrieve the section.

Claim 7 also requires that the server retrieve the section from the mailfile and create an object having the section expanded in the retrieved document, and the server removes the URL from the retrieved document.

As best can be determined, the Examiner cites Grout as

teaching these inserting, retrieving, and removing steps of claim 7. Specifically, the Examiner cites Grout column 1, lines 1 - 46; column 3, lines 13 - 32, column 5, lines 45 - 61; column 6, lines 60 - 63; column 7, lines 20 - 44; and FIG. 2A. Appellants contend that the cited portions of Grout do not teach these steps of claim 7.

The inserting, retrieving, and removing steps of claim 7 clearly must be performed at the server. Grout, however, describes in column 1, lines 1 - 46, a client browser requesting a document by its URL. Appellants' claim 7 requires the requesting application to be running on a user workstation different from the server. Grout's client browser then receives the document which may itself have embedded URL's linking to objects, and the client browser, not the server, downloads the linked objects for presentation. All of the action in Grout column 1, lines 1 - 46, is being performed at the client browser whereas Appellants' claim 7 specifically requires the steps be performed at the server. The requesting application, e.g., client browser, must be running on a user workstation different from the server. Therefore, Grout is not describing or teaching steps being performed at the server.

FIG. 2A of Grout merely shows a document having linked objects at a client computer. Some of the linked objects are stored on the client computer so that the client browser does not have to request these from a server over a network, thus reducing network traffic. There is no description in FIG. 2A of the important steps just mentioned which according to claim 7 must be performed by the server.

Furthermore, the cited portions of columns 3, 5, 6, and 7 also do not describe or suggest these important steps which must be performed by the server. Nor does Benninghoff nor Don describe or suggest these steps. Claim 7 is therefore allowable. Likewise, independent claim 16 is allowable for the same reasons. All of the remaining claims depend directly or indirectly on allowable claim 7 or allowable claim 16 and are therefore also allowable.

Appellants' position in view of the above is that the first ground of rejection of pending claims 7 - 10 and 12 - 19 is in error and must be withdrawn. All of the claims are allowable under 35 U.S.C. 103(a) over Benninghoff, Grout, and Don.

Second Ground of Rejection:

As noted above, Appellants do not argue the allowability of claim 11 separately. Claim 11 is allowable based on its dependence on allowable claim 7 as argued above. The rejection of claim 11 under 35 U.S.C. 103(a) over Benninghoff, Don, Grout, and Little is therefore moot and must be withdrawn.

In view of the above, Appellants respectfully request that the Board reverse the Examiner's final rejection of all of the claims on appeal, and allow these claims.

Respectfully submitted,

Dated: 12/27/2006

By: /John Pivnichny/
John R. Pivnichny
Reg. No. 43,001

Telephone: (607)429-4358

Fax: (607)429-4119

(viii) CLAIMS APPENDIX

Claims 7 - 19 are pending.

7. (Previously Amended) A method of providing data to an application, comprising the steps of:

providing a mailfile stored on a server, of documents having a section and fields;

receiving a request as a SOAP protocol message from an application running on a user workstation different from said server, for one of said documents;

retrieving said fields of said one of said documents from said mailfile;

in response to said fields, retrieving said one of said documents as a markup language document;

inserting at said server, a URL into said markup language document to retrieve said section of said one of said documents;

retrieving at said server, said section from said mailfile in said markup language;

removing at said server, said URL from the retrieved document and creating an object having said section expanded in the retrieved document; and

marshalling said object and sending the marshalled object to said application as a SOAP protocol message.

8. (Original) The method of claim 7, wherein said fields are retrieved as an XML document.

9. (Original) The method of claim 7, wherein said markup language is HTML or XHTML.

10. (Original) The method of claim 7, wherein said one of said documents has a file attachment link.

11. (Original) The method of claim 10, further comprising the steps of retrieving said attachment, removing said link, and inserting said attachment into said object.

12. (Original) The method of claim 7, wherein said one of said documents has an image tag.

13. (Original) The method of claim 12, further comprising the steps of retrieving the image of said image tag, encoding said image, and inserting the encoded image in place of said image tag in the retrieved document.

14. (Original) The method of claim 7, wherein said one of said documents has a link to other items in said document.

15. (Original) The method of claim 14, further comprising the steps of retrieving the content of said link, and inserting said content in the retrieved document at the position of said link.

16. (Previously Amended) A system for providing data stored in a file to an application, comprising:

a file stored on a server, having data stored as documents having sections;

a database on said server, for passing a request for one of said documents to said file and upon return converting said one of said documents into an extended markup format;

an authentication directory having authentication records for an application;

web service software for receiving a request from an application running on a user workstation different from said server, for one of said documents, retrieving said one of said documents, and creating an extended markup object containing said document;

web service software for inserting at said server, a URL into said markup object to retrieve one of said sections;

web service software for retrieving at said server, said one of said sections from said file;

web service software for recovering at said server, said URL from the retrieved one of said documents and creating an object having said one of said sections expanded in the retrieved document; and

a protocol tool for authenticating said application using said records, marshalling said object, and sending the marshalled object in a simple object access protocol to said application.

17. (Original) The system of claim 16, wherein said software and said tool are adapted to operate without the need for a mail or calendaring client.

18. (Original) The system of claim 16, wherein said extended markup format is XML.

19. (Original) The system of claim 16, wherein said object is marshalled into said simple object access protocol according to a pre-defined set of rules.

(ix) EVIDENCE APPENDIX

None.

(x) RELATED PROCEEDINGS APPENDIX

No decision has been rendered by a court or the Board in the related appeal identified above in section (ii).